NASA/TM-97-206287

A Web-Based System for Monitoring and Controlling Multidisciplinary Design Projects

Andrea O. Salas and James L. Rogers
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

December 1997

Available from the following:

NASA Center for AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

A Web-Based System for Monitoring and Controlling Multidisciplinary Design Projects

Andrea O. Salas and James L. Rogers
NASA Langley Research Center

Abstract

In today's competitive environment, both industry and government agencies are under enormous pressure to reduce the time and cost of multidisciplinary design projects. A number of frameworks have been introduced to assist in this process by facilitating the integration of and communication among diverse disciplinary codes. An examination of current frameworks reveals weaknesses in various areas such as sequencing, displaying, monitoring, and controlling the design process. The objective of this research is to explore how Web technology, in conjunction with an existing framework, can improve these areas of weakness. This paper describes a system that executes a sequence of programs, monitors and controls the design process through a Web-based interface, and visualizes intermediate and final results through the use of Java™* applets. A small sample problem, which includes nine processes with two analysis programs that are coupled to an optimizer, is used to demonstrate the feasibility of this approach.

1. Introduction

The World Wide Web is a method for distributing information to a widely distributed user base. Information is accessed through Web pages that are viewed with Web browsers, such as Netscape Navigator, Mosaic, or Microsoft Internet Explorer. Because Web pages are becoming much more interactive, new tools and techniques are needed to exploit this capability. The purpose of this research is to explore the application of Web technology to the monitoring and controlling of multidisciplinary design projects. A multidisciplinary design project involves the integration of various discipline codes that may have been developed on different computer architectures and that must be executed on heterogeneous computers distributed over a network.

Web technology has much to offer in monitoring and controlling design projects. Web browsers offer convenient access to information that resides on remote computers. Web software is already available to most researchers, and many users are already familiar with this interface. Because the Web promotes communication among codes and computers, it can be used to reduce design

* Java is a trademark of Sun Microsystems, Inc.

cycle time and costs. The reduction of design cycle time is a critical goal for both industry and government agencies.

For the current research, a Web-based system has been designed to assist in monitoring and controlling multidisciplinary design projects. The use of this system is based on the assumption that project team members create Web pages that describe various processes of the project, such as discipline analyses and optimization. A project manager should be responsible for developing a main project Web page with links to other process pages. These Web pages may provide links to information (e.g., data, documentation) which is associated with individual processes.

After all processes have been defined for a design project, the project team members identify the couplings among the processes. The sequence for executing these processes can then be derived from knowledge regarding the processes and their couplings. Note that in most multidisciplinary design projects, the design cycle involves iterations that must be taken into account in sequencing and executing the processes.

After the sequencing for the processes has been determined, the system executes the processes in the appropriate order. A Web page created and updated by the system is used to monitor the status of the design cycle. This Web page also contains links to the user-defined Web pages and process control options. Process control options, such as disabling a process or manually invoking a process, can be used to alter the process execution sequence. A main control program collects control requests and uses them to modify the current set of processes that are ready for execution. The user-defined Web pages, associated with particular processes, may provide additional monitoring capabilities.

This paper contains a brief discussion of the current framework development efforts, particularly in the areas of sequencing and execution of design processes. Next, sequencing and display tools are described. Then, a Web-based system is described in detail, and the feasibility of this system is demonstrated with a small multidisciplinary problem that consists of two analyses coupled to an optimizer. This paper concludes with plans for future study and comments on the current research.

2. Execution Control in Multidisciplinary Frameworks

For the purpose of this report, a framework for multidisciplinary computational environments is defined as a hardware and software architecture that enables integration, execution, and communication among diverse disciplinary processes. A great deal of effort is being expended in the U.S. and abroad by government agencies, industry, and universities in the development of multidisciplinary frameworks (refs. 1-10). In these frameworks, process execution can be handled in several ways:

(a) A master program, written in a compiled language such as C or C++, can be used to code a fixed execution sequence of processes. Processes may be executed through subroutine or system calls.

(b) A master program, written in an interpreted or script language such as tcl, can also be used to code a fixed execution sequence of processes.

(c) Graphical methods can be used to link the processes and to provide more flexibility in the sequence of their execution. The underlying code actually establishes the links.

Through personal communication, the authors have determined that most framework developers are exploring methods for incorporating Web technology into their framework development efforts; however, these efforts have not been discussed in the literature.

The method presented here is developed around a knowledge-based system made up of a problem-independent set of rules that determine the processes ready for execution and a problem-dependent set of facts that describe the individual processes and their couplings. Thus, this system provides a dynamic and flexible method for executing the processes found in a multidisciplinary problem. The system is further enhanced by introducing Web technology, which provides the mechanisms for interactive alteration of sequencing control and execution monitoring. The system capabilities were coded to facilitate integration into any framework, regardless of the current sequencing and execution methods of the framework.

3. Process Sequencing

After all of the processes and couplings in a multidisciplinary project are identified, the optimum sequence for executing the processes must be determined. One tool for determining this sequence is the Design Manager's Aid for Intelligent Decomposition, otherwise known as DeMAID (ref. 11); DeMAID is a knowledge-based software tool that can be used to assist a project manager in making decisions that can potentially reduce the time and cost of a design cycle. The DeMAID software minimizes the number of feedback couplings and groups processes into iterative subcycles. After the iterative subcycles have been determined, DeMAID sequences the processes within each subcycle with a genetic algorithm in order to minimize the time and cost of the design cycle (ref. 12). Coupling strengths are used to estimate the number of iterations required for convergence (ref. 13).

4. Displaying Sequence of Processes

After the iterative subcycles and the optimum sequence for execution have been determined, DeMAID displays the sequence in a design structure matrix (DSM) format, which was originally formulated by Steward (ref. 14). A sample DSM is shown in figure 1.

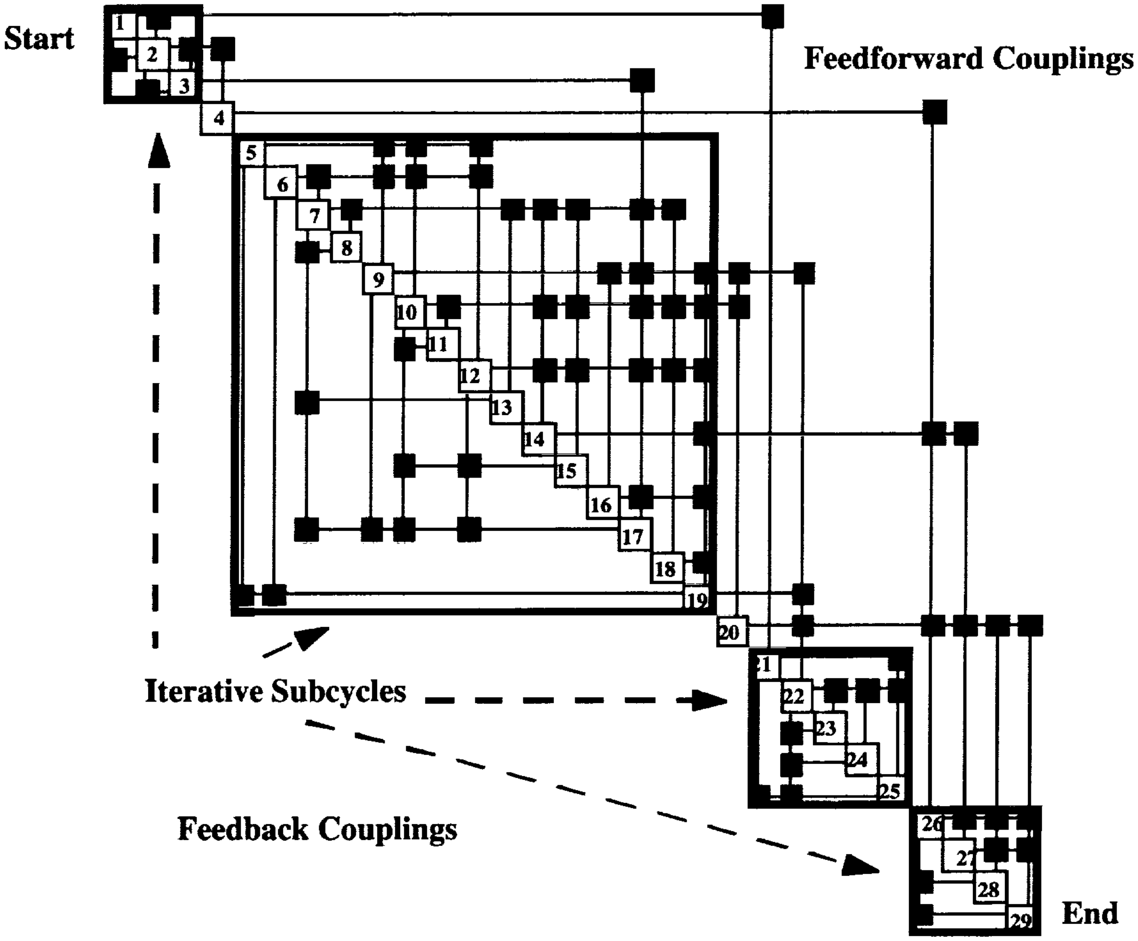


Figure 1. Sample DSM.

The DSM in figure 1 shows the design processes as numbered boxes on the diagonal. Output from a process is shown as a horizontal line that exits a numbered box, and input to a process is shown as a vertical line that enters a box. The off-diagonal squares that connect the horizontal and vertical lines represent couplings between two processes. Squares in the upper triangle of the DSM represent feedforward couplings; squares in the lower triangle of the matrix represent feedback couplings. Feedback couplings imply iterations, and initial data must be estimated. Feedback couplings should be eliminated where possible to reduce the number of iterative cycles. If certain feedback couplings cannot be eliminated, then the processes are grouped into iterative subcycles.

In figure 1, processes 1-3, 5-19, 21-25, and 26-29 are grouped into iterative subcycles. The primary advantage of the DSM format over earlier display tools (e.g., Program Evaluation Review Technique (PERT) charts) is the capability to group and display these iterative subcycles, which are common to design projects.

5. Web-Based System

The purpose of the Web-based system described in this paper is to provide the end user with methods for sequencing, displaying, monitoring, and controlling the processes that make up a multidisciplinary design project. The monitoring and control system consists of several parts, as shown in figure 2.

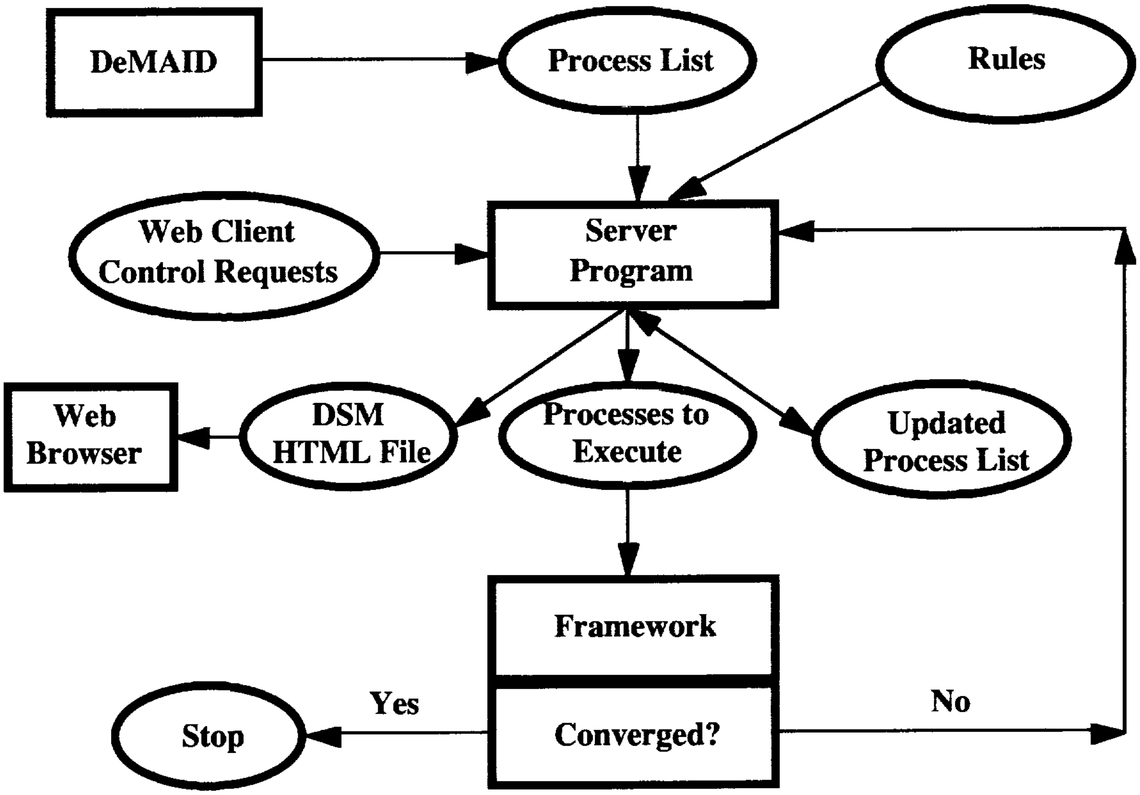


Figure 2. Monitoring and control system.

5.1 System Features

For this system, the main program that facilitates monitoring and control is referred to as the "server program." Input to the server program consists of (1) a file output by DeMAID that contains a problem-dependent list of sequenced processes and the links to the process Web pages, (2) a file with a set of problem-independent rules that may be applied to any design project, and (3) Web client control requests. The server program, a mixture of C and Java code (ref. 15), produces three output files: an updated file that lists the processes, a file that lists those processes that are ready for execution, and a HyperText Markup Language (HTML) file for creating a DSM with the latest status of the design project.

To produce these files, the server program uses various Java language features, such as threads, sockets, and synchronized access to object functions. The server program accepts Web-client control requests and records them. In addition, the server program calls the CLIPS knowledge-base system (ref. 16) and then executes the framework code.

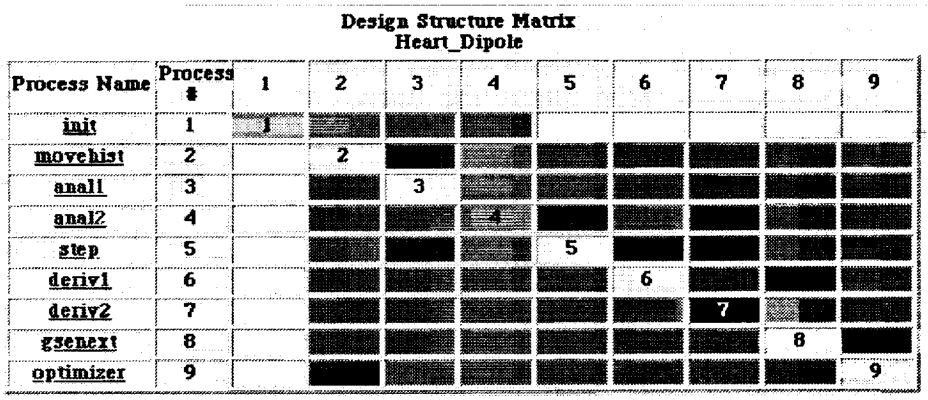
Each time CLIPS is called, it loads the fact list, which contains process and Web data, a rule file, and the current set of control requests. The rules contain knowledge for determining which processes are ready for execution based on the satisfaction of their input requirements; for creating a file that contains the names of those processes that are ready for execution, along with paths to the locations of their input, output, and executable files; for creating the HTML file for displaying the DSM as a Web page; and for updating the status of the processes that are contained in the facts list.

While the concepts used here are applicable to any multidisciplinary framework, a sample framework is written in C. This framework reads the file that contains the list of those processes that are ready for execution (which is output by the server program) and executes these programs sequentially. The first pass through the system assumes that all data that are represented by feedback couplings are available.

5.2 Monitoring and Control Features

After the main project Web page has been downloaded, the project manager executes the server program to control the project and monitor the status of various processes of the design project. The DSM is displayed from the main project page and is described with HTML table tags (fig. 3). Table cells on the diagonal correspond to processes; those on the off-diagonal correspond to couplings among processes. A <META> tag in the HTML file automatically updates the DSM Web page every few seconds; by using different colors, the DSM displays the current status information for the design project. In particular, colors indicate the following: green indicates processes that are executing; yellow, processes that are waiting for data; black, processes that are temporarily deactivated; turquoise, processes that will not execute again; blue, data are available; red, data are not available; and gray, an iterative subcycle. These colors change as different processes are executed. The output couplings for processes that are either turquoise or black remain blue to indicate that the data for these processes are always available.

The HTML page in Figure 3 is a snapshot from the sample problem execution; this sample problem is described later in the paper. The snapshot reveals the following: process 1 will not run again; process 4 is ready for execution; process 7 has been deactivated; and all other processes are waiting for data. In the color version of this figure, the process colors are as follows: process 1 is turquoise; processes 2, 3, 5, 6, 8, and 9 are yellow; process 4 is green; and process 7 is black. The colors of the off-diagonal cells reveal that only process 4 has all data available, that is, all coupling cells in column 4 of the matrix are blue. Coupling cells where data is not available are red; while non-coupling cells are either gray, indicating an iterative subcycle , or white.



Design Structure Matrix
Heart_Dipole

Process Name	Process #	1	2	3	4	5	6	7	8	9
init	1	1	■	■	■					
movehist	2		2	■	■	■	■	■	■	■
anal1	3		■	3	■	■	■	■	■	■
anal2	4		■	■	4	■	■	■	■	■
step	5		■	■	■	5	■	■	■	■
deriv1	6		■	■	■	■	6	■	■	■
deriv2	7		■	■	■	■	■	7	■	■
gsenext	8		■	■	■	■	■	■	8	■
optimizer	9		■	■	■	■	■	■	■	9

Figure 3. Sample DSM in HTML format.

Java applets are implemented to display current and historical data and are invoked from within the HTML process pages. These applets can show the history of the objective function, selected responses, and design-variable values, as well as the status of the constraints. The applets reference the location of the current set of data to be processed; the locations are defined as parameters in the HTML pages.

Control options can also be set by using HTML forms. From the DSM, the user may click on the process name to display the current setting for the control option that needs to be modified. Changes can then be made and submitted. Currently, the only available option is to deactivate or activate a process. A process may be deactivated if the output from the process is not changing. Other potentially useful control options include the option to manually or automatically invoke certain processes, the option to change coupling strengths, and the option to pause or restart execution.

6. Sample Problem

The heart dipole problem from the NASA Langley Research Center Multidisciplinary Optimization Branch Test Suite (ref. 17) was selected to demonstrate the Web-based monitoring and control system. This test suite was created to demonstrate multidisciplinary design optimization (MDO) solution methods. The heart dipole problem arises from the experimental electrolytic determination of the resultant dipole moment in the heart. The conventional solution method is to solve a set of nonlinear equations with eight unknowns. The problem is recast as an optimization problem and consists of nine processes, including two analysis programs and an optimizer (ref. 18).

The DeMAID software was applied to sequence the processes for this problem. The resulting DSM is shown in figure 4, and the HTML representation is shown in figure 3. As shown in figure 4, this problem contains two iterative cycles, identified by the feedback coupling from process 5 to process 3 and the feedback coupling from process 9 to process 2. The smaller cycle, which contains processes 3-5, represents the iterations required to converge data from two analysis programs. The larger cycle, which contains processes 2-9, represents the overall optimization cycle in which the design variables, constraints, and an objective function are input to the optimizer (process 9) and a new set of design variables is computed and fed back to the analysis programs. Process 1, which is the initialization process, executes only once.

###	Label
1	init
2	movehist
3	anal1
4	anal2
5	step
6	deriv1
7	deriv2
8	gsenext
9	optimizer

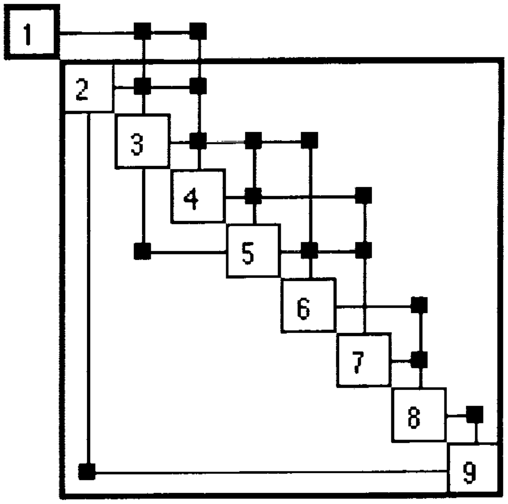


Figure 4. Sample DSM for heart dipole problem.

The number of iterations in a cycle is determined by the coupling strength of the feedback. Seven levels of coupling strength are specified in DeMAID. These strengths range from extremely weak to extremely strong, with the middle level, nominal, as the default. The system assumes that extremely strong couplings require eight iterations to converge and extremely weak couplings require two iterations. Therefore, as the coupling strength of the feedback from process 5 to process 3 is very weak, the system will execute processes 3, 4, and 5 three times before proceeding to execute other processes.

To integrate this problem into the Web-based system, the optimization module was modified to maintain a history for each design variable and for the objective function in separate files. This modification is not unreasonable because easy access to this data can be useful for later viewing. The initial values at the start of the design cycle and the values at the end of each iteration are recorded. In addition, the current constraint values at the end of each iteration are written to a file.

Several links may be chosen from the optimization module Web page, which should be accessible from the main Web page for the project. One link downloads a Web page that contains an applet for displaying the history of the objective function. The user can click the applet Update button to update the objective function at any time. Alternatively, the user can specify that the applet automatically update the graph at regular intervals. The history of the objective function for the sample problem after nine iterations is displayed in figure 5.

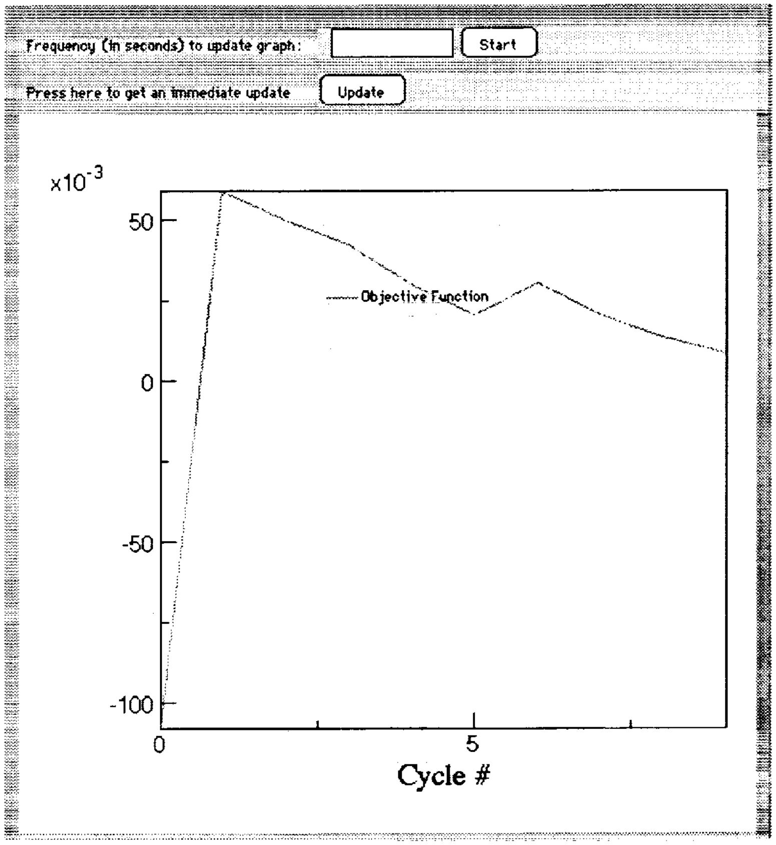

Frequency (in seconds) to update graph:
Start
Press here to get an immediate update
Update
x10^{-3}
50
0
-50
-100
0
5
Cycle #
Objective Function

Figure 5. Applet for monitoring objective function.

The user can click on the remaining optimization Web-page link to download an HTML page that contains two applets: one for monitoring the design variables and one for monitoring the constraints. The design-variable monitoring applet displays the list of design variables. From this list, the user can select a design variable and produce a history of its changes. For example, the x(3) variable was selected from the sample problem after nine iterations to produce the history shown in figure 6. The sample problem contains four constraints, for which current values are recorded after each iteration. The applet that monitors the constraint status displays the status of each constraint by using a color code: red indicates a violated constraint; green indicates an active constraint; and yellow indicates a feasible, inactive constraint.

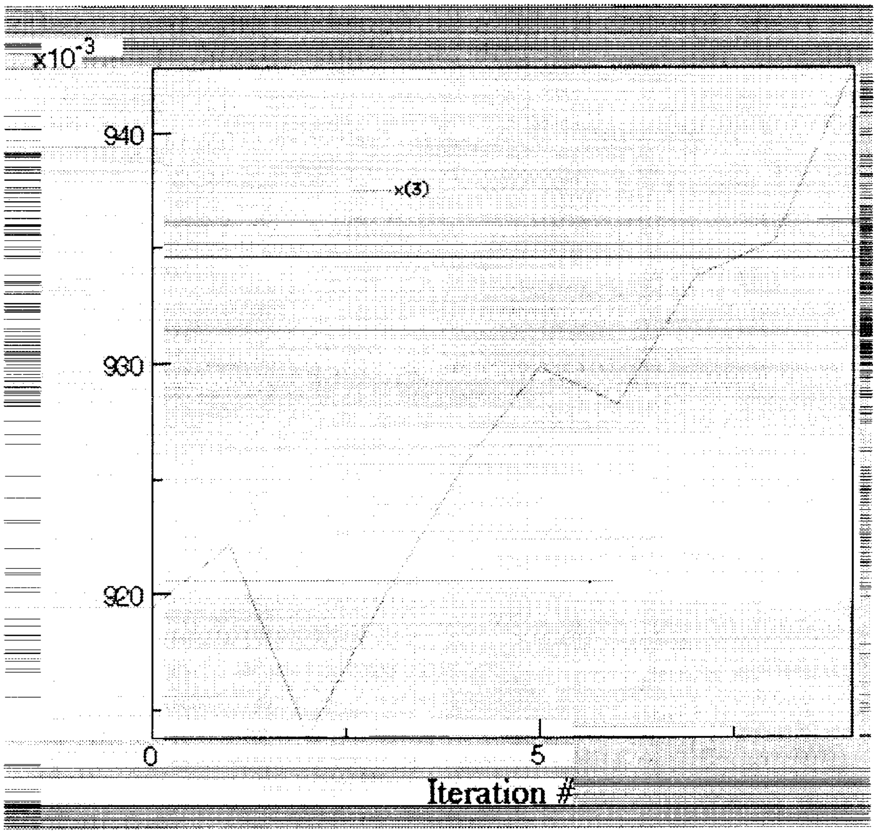


Figure 6. History of design variable x(3).

7. Future Plans

Future work may concentrate in several areas. One area of focus will be to demonstrate the Web-based system in a more realistic setting. For the demonstration discussed in this paper, the processes and the files that were associated with the sample program resided in one directory and were executed on the same machine. A more realistic scenario would require that processes and associated data be located in various directories, possibly on different file systems, and that the processes be executed on different machines. Another area of focus will be to test these concepts on a problem with an existing framework, such as FIDO (ref. 1) or iSIGHT (ref. 7).

The monitoring and control features of this system also may be expanded. The monitoring features can be enhanced to include the visualization of analysis data that can be accessed via the process Web pages. Applets similar to those used here for the sample optimization process can be used to collect data for selected analysis responses. Furthermore, several new control options also may be added. For example, one new control feature that is being considered is the option to manually invoke a process. Another option that is under consideration is to allow the user to select a specific application to be executed for each process. This option may be desirable if varying levels of fidelity for a particular process (discipline) are to be used.

8. Concluding Remarks

In this paper, a flexible approach for monitoring and controlling multidisciplinary design optimization projects has been described. This approach uses a combination of World Wide Web technology and knowledge-base methods. Given both a description of the processes for a particular design project and the sequencing information, the knowledge base applies a set of problem-independent rules to determine a set of processes that are ready for execution. These processes are executed by an independent framework.

A combination of Web programming techniques provides the means both for displaying the status of the execution and design optimization results and for interfacing with Web pages to control the execution. Web technology provides a convenient and familiar interface for accessing data that are produced from independent processes and for interfacing with the design project.

In summary, several steps must be taken by a project team in order to use the Web-based system described in this paper. Integration with the knowledge base system requires the following activities: 1) major processes of the multidisciplinary project must be identified; 2) the process coupling information must be defined and used by DeMAID to generate the sequencing information for execution; and 3) a master code or framework that invokes the various processes must be provided (and possibly modified) to accept the list of processes identified as ready for execution by the knowledge-based system. Integration with the Web tools requires: creation of a project Web page that contains the links to the project design structure matrix (DSM) and process Web pages and creation of process Web pages that contain links to information that describes the process and to data that can be used by generic applets to display design information.

9. References

1. Weston, R. P.; Townsend, J. C.; Eidson, T. M.; and Gates, R. L.: A Distributed Computing Environment for Multidisciplinary Design. AIAA Paper No. 94-4372, September 1994.

2. Eldred, M. S.; Outka, W. J.; Bohnhoff, W. J.; Witkowski, V. J.; Ponslet, E. R., and Chen; K. S.: Optimization of Complex Mechanics Simulations with Object-Oriented Software Design. AIAA Paper No. 95-1433, April 1995.

3. Hufford, G. S.; Kingsley, G. M.; and Harrand, V. J.: Requirements, Preliminary Design and Initial Implementation of a Generic Multi-Disciplinary Software Environment. AIAA Paper No. 97-0929, January 1997.

4. Blair, M.; Bharatram, G.; Canfield, R. A.: Designing a Blended Composite Wing and Fuselage. AIAA Paper No. 96-3995, September 1996.

5. Hale, M. A.; Craig, J. I.; Mistree, F.; and Schrage, D. P.: DREAMS & IMAGE: A Model and Computer Implementation for Concurrent, Life-Cycle Design of Complex Systems. Concurrent Engineering: Research and Applications, vol. 4, no. 2, June 1996, pp. 171-186.

6. Multidisciplinary Integrated Design Assistant for Spacecrafts (MIDAS) Tutorial, Jet Propulsion Laboratory, November 10, 1995.

7. iSIGHT 2.0, Engineous Software Inc., 1996.

8. Ridlon, S. A.: A Software Framework for Enabling Multidisciplinary Analysis and Optimization. AIAA Paper No. 96-4133, September 1996.

9. Daum, A.; and Wolf, D. M.: TRANSYS--A Multidisciplinary Software System for Preliminary Design, Analysis, and Evaluation of Space Transportation Systems. AIAA Paper No. 94-4342, September 1994.

10. Rahn, M.; Schottle, U. M.; and Messerschmid, E.: Multidisciplinary Design Tool for System and Mission Optimization of Launch Vehicles. AIAA Paper No. 96-4130, September 1996.

11. Rogers, J. L.: DeMAID/GA-An Enhanced Design Manager's Aid for Intelligent Decomposition. AIAA Paper No. 96-4157, September 1996.

12. Rogers, J. L.; McCulley, C. M.; and Bloebaum, C. L.: Integrating a Genetic Algorithm into a Knowledge-Based System for Ordering Complex Design Processes, Proceedings of the Artificial Intelligence in Design '96 Conference, June 1996, pp. 119-133.

13. Rogers, J. L.; and Bloebaum, C. L.: Ordering Design Tasks Based on Coupling Strengths. AIAA Paper No. 94-4326, September, 1994.

14. Steward, D. V.: Systems Analysis and Management, Structure, Strategy and Design. Petrocelli Books, Inc., 1981.

15. Lemay, L.; and Perkins, C. L.: Teach Yourself Java™ in 21 Days. Sams.net Publishings, 1996.

16. Giarratano J.; and Riley, G.: Expert Systems Principles and Programming, Kent Publishing Company, 1989.

17. Padula, S. L.; Alexandrov, N.; and Green, L. L.: MDO Test Suite at NASA Langley Research Center. AIAA Paper No. 96-4028, September 1996.

18. Vanderplaats, G. N.: CONMIN--A FORTRAN Program for Constrained Function Minimization: User's Manual. NASA TMX-62282, August 1973.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
December 1997

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
A Web-Based System for Monitoring and Controlling Multidisciplinary Design Projects

5. FUNDING NUMBERS
509-10-11-01

R23109

6. AUTHOR(S)
Andrea O. Salas and James L. Rogers

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-17685

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM-97-206287

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 61 Distribution: Nonstandard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

In today's competitive environment, both industry and government agencies are under enormous pressure to reduce the time and cost of multidisciplinary design projects. A number of frameworks have been introduced to assist in this process by facilitating the integration of and communication among diverse disciplinary codes. An examination of current frameworks reveals weaknesses in various areas such as sequencing, displaying, monitoring, and controlling the design process. The objective of this research is to explore how Web technology, in conjunction with an existing framework, can improve these areas of weakness. This paper describes a system that executes a sequence of programs, monitors and controls the design process through a Web-based interface, and visualizes intermediate and final results through the use of Java™ applets. A small sample problem, which includes nine processes with two analysis programs that are coupled to an optimizer, is used to demonstrate the feasibility of this approach.

14. SUBJECT TERMS
MDO, multidisciplinary design optimization, MDO frameworks, World Wide Web, DeMAID, process management

15. NUMBER OF PAGES
18

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT